



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/681,567	04/30/2001	Walter Dixon III	345708002US	9844

25096 7590 11/05/2003

PERKINS COIE LLP  
PATENT-SEA  
P.O. BOX 1247  
SEATTLE, WA 98111-1247

EXAMINER
----------

GROSS, KENNETH A

ART UNIT	PAPER NUMBER
2122	

DATE MAILED: 11/05/2003

Please find below and/or attached an Office communication concerning this application or proceeding.

# Office Action Summary

Application No.

09/681,567

Applicant(s)

DIXON ET AL.

Examiner

Kenneth A Gross

Art Unit

2122

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

## Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

## Status

- 1) ☐ Responsive to communication(s) filed on \_\_\_\_.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

## Disposition of Claims

- 4) ☒ Claim(s) 1-57 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 1-57 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_ are subject to restriction and/or election requirement.

## Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- 11) ☐ The proposed drawing correction filed on \_\_\_\_ is: a) ☐ approved b) ☐ disapproved by the Examiner.  
If approved, corrected drawings are required in reply to this Office action.
- 12) ☐ The oath or declaration is objected to by the Examiner.

## Priority under 35 U.S.C. §§ 119 and 120

- 13) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).  
a) ☐ All b) ☐ Some \* c) ☐ None of:  
1. ☐ Certified copies of the priority documents have been received.  
2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_.  
3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).  
\* See the attached detailed Office action for a list of the certified copies not received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application).  
a) ☐ The translation of the foreign language provisional application has been received.
- 15) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121.

## Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892) 4) ☐ Interview Summary (PTO-413) Paper No(s). \_\_\_\_
- 2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948) 5) ☐ Notice of Informal Patent Application (PTO-152)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) \_\_\_\_ 6) ☐ Other: \_\_\_\_

## **DETAILED ACTION**

### ***Claim Objections***

1. Claims 18 are objected to because of the following informalities: Claim 18 states “a translator for preparing command” (Page 29, lines 2-3). This should be “a translator for preparing a command”. Claim 55 states “the commands of an interactions” (Page 33, lines 23). This should be “the commands of an interaction[s]”. Appropriate correction is required.

### ***Claim Rejections - 35 USC § 112***

2. The following is a quotation of the second paragraph of 35 U.S.C. 112:

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.

3. Claims 1-20, 42, 47, 56, and 57 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Specifically, Claim 1 states “a command having an attribute” which is unclear (Page 27, line 5). Claim 1 further teaches, “input attributes of the command” and “output attributes of the command”. When Claim 1 first teaches “a command having an attribute”, is this an input or an output attribute? Furthermore, if the command has input and output attributes, doesn't the term “a command having an attribute” indicate only one attribute? Claim 7 recites the term “performing of the view”. What does it mean exactly to perform a view? Claims 56 and 57 are rejected for the same reasons as Claim 7. Claim 42 recites the limitation

"the behavior" in Page 32, line 10). There is insufficient antecedent basis for this limitation in the claim. Claims 2-6, 8-20, and 43-47 are rejected for being dependent on a rejected base claim.

***Claim Rejections - 35 USC § 102***

4. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

5. Claims 33-35 are rejected under 35 U.S.C. 102(e) as being anticipated by Limondin et al. (U.S. Patent Number 6,226,783).

In regard to Claim 33, Limondin teaches a data structure including a command descriptor (Figure 5, item 200) identifying an object class (Column 4, lines 35-37), the object class defining input and output variable (Figure 5, items 510 and 234) and having a perform method (Figure 7, item 650). The program is executed by instantiating objects and setting input attribute values of objects based on output values of objects (Column 4, lines 42-45) and running the perform method (Column 3, lines 3-4).

In regard to Claim 34, Limondin teaches that each step has a list of commands associated with it, where steps are operations that interact with graphics (Column 3, lines 3-4).

In regard to Claim 35, step operations are grouped into interaction step objects, which inherently have a name (Column 4, lines 35-37).

6. Claim 48 is rejected under 35 U.S.C. 102(b) as being anticipated by Silberbauer et al. (U.S. Patent Number 5,355,493).

In regard to Claim 48, Silberbauer teaches: (a) identifying input and output attributes of the command (Column 4, line 23 and lines 43-44); (b) setting a value for each identified input attribute by retrieving from an attribute store (Column 4, lines 24-25 and lines 37-41); (c) performing the behavior of the command (Column 4, line 42); (d) retrieving a value for an identified output attribute, and (e) storing the retrieved value in the attribute store (43-46).

***Claim Rejections - 35 USC § 103***

7. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

8. Claims 1-5, 15, 16, 19, 20, 42-44, 56, and 57 are rejected under 35 U.S.C. 103(a) as being unpatentable over "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493).

In regard to Claim 1, Perry teaches: providing a plurality of definitions of interactions of the program in the form of Visual Basic buttons that allows a user to interact with a program (Page 4, lines 7-10); each interaction having one or more command definitions in the form of Visual Basic code executed in response to the interaction (Page 5, lines 1-4); a view definition in the form of a Visual Basic Window or Dialog Box (Page 3, Figure 1.8); a command having an attribute and a behavior. Perry teaches the concept of a function which takes arguments

Art Unit: 2122

(attributes) and performs a behavior based on the function code (Page 19); receiving a request by way of an event which lets the program know that a user requests some function to be performed (Page 3, lines 11-12); identifying the interaction associated with the received request by recognizing the user triggered event and performing the associated event procedure (Page 5, lines 6-9); and providing a response generated by the view associated with the identified interaction.

Perry further teaches for each command of the identified interaction, performing the behavior of the command with input attributes and extracting attributed values of the command by retrieving values of the output attributes of the command. Perry does not teach preparing the command by setting values of input attributes of the command based on attribute values stored in an attribute store and storing the out attributes in an attribute store. Silberbauer, however, does teach storing attribute values (Column 4, lines 24-25), setting values of a function input attributes (Column 4, lines 37-41), and storing output attribute values in an attribute store (Column 4, lines 43-49).

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to define interactions of a program with command definitions associated with it where command definitions contain an attribute and a behavior, a view definition, where a request of an interaction is received and the interaction identified, performing the behavior of each command of the interaction, and receiving output variables of the command, and providing a response generated by the view associated with the interaction, as taught by Perry, where the command is prepared by setting values of input attributes of the command based on attribute values stored in an attribute store and storing the out attributes in an attribute store, as taught by Silberbauer, since this allows variables to be saved and accessed at later times.

In regard to Claim 2, Silberbauer teaches a data stream request to invoke a function, where the attribute values are stored in the request (Column 4, lines 20-25).

In regard to Claim 3, Perry teaches validating values of attributes by means of validation functions before performing a behavior on the attributes (Page 33, lines 6-8).

In regard to Claim 4, the examiner takes official notice that an assignment statement in command code assigns values to attributes.

In regard to Claim 5, Perry teaches defining an attributes scope (Page 22, lines 31-33).

In regard to claim 15, Perry teaches the concept of Case..Else statements, which are conditional statements that specifies multiple commands to be performed based on the value of an attribute (Page 52, lines 1-3).

In regard to Claim 16, Silberbauer teaches placing values on a function's input parameters, where certain values go to certain parameters (Column 4, lines 35-41). Hence, the parameters stored in the data stream must map to parameters of the command.

In regard to Claim 19, the examiner takes official notice that a translator function is a well-known function for translating source code into object code, and it would be obvious for the command function to identify a translator, since this would allow the command function to operate as a translator.

In regard to Claim 20, the examiner takes official notice that an assignment statement in command code assigns values to attributes, and it would be obvious to place an assignment statement in a command to overwrite the value of an attribute.

In regard to Claim 42, Perry teaches receiving a request by way of an event which lets the program know that a user requests some function to be performed (Page 3, lines 11-12);

Art Unit: 2122

identifying the interaction associated with the received request by recognizing the user triggered event and performing the associated event procedure (Page 5, lines 6-9). Perry further teaches for each command of the identified interaction, performing the behavior of the command with input attributes and extracting attributed values of the command by retrieving values of the output attributes of the command. Perry does not teach preparing the command by setting values of input attributes of the command. Silberbauer, however, does teach storing attribute values (Column 4, lines 24-25) and setting values of a function input attributes (Column 4, lines 37-41). Silberbauer further teaches a system for carrying out the above method. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build a system for receiving a request to perform an interaction of the application, means for performing the behavior associated with the commands, and receiving output attribute values of commands after performing the behavior of the command, as taught by Perry, where each command is prepared with input values of the attributes of the command, since this allows the function to be run with given values, without need for the function to find the values during execution.

In regard to Claim 43, Silberbauer teaches storing output variables in an attribute store (Column 4, lines 43-49).

In regard to Claim 44, Silberbauer does teach retrieving stored values from an attribute store (Column 4, lines 24-25).

In regard to Claim 56, Perry teaches providing a plurality of interactions in the form of events (Page 4), where an interaction can have an associated command, which executes when the interaction occurs (Page 5, lines 1-4), such as a button on a dialog box that performs some function. If the user chooses not to write any code for a given event, the event can still have a



Art Unit: 2122

view, such as a button on a dialog box, but no command associated with the event, if the button is pressed (Page 5, lines 5-7). Perry teaches receiving an indication of an interaction of a program (Page 5, lines 5-6), performing the behavior associated with the command (Page 5, lines 1-4), and when the interaction has a view performing the view, such as displaying results to the screen after a user presses a button to perform a calculation.

In regard to Claim 57, Perry teaches altering a command button in the visual interface, hence determining in which way to view the button (Page 4, lines 10-15).

9. Claims 8-11 and 14 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Laskoski (U.S. Patent Number 5,428,554).

In regard to Claim 8, Perry and Silberbauer teach the method of Claim 1, but do not teach that the view definition defines a view with attributes and a behavior, and performing the view includes setting values of input attributes of the view based on attribute values stored in attribute store, and performing the behavior of the view with the attributes. Laskoski, however, does teach a display function that performs a display behavior and takes a display threshold attribute (Column 4, lines 12-14). Once the attribute is set, the display function performs the display behavior and displays data on the screen. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1 where the view definition defines a view with attributes and a behavior, and performing the view includes setting values of input attributes of the view based on attribute values stored in attribute store, and performing the behavior of the view with the attributes, as taught by Laskoski, since this

Art Unit: 2122

allows output of a command to be displayed on a computer screen so that a user can view output data.

In regard to Claim 9, the examiner takes official notice that a page is a well-known display means for displaying data on a computer.

In regard to Claim 10, Perry and Silberbauer teach the method of Claim 1, but do not teach that the view definition defines a view with a target and including invoking the target to generate a response. Laskoski, however, teaches displaying data on a target device (Column 2, lines 37-39). The device generates a response by displaying the data on the display means of the device. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the view definition defines a view with a target and including invoking the target to generate a response, as taught by Laskoski, since this allows the display function to generate a display on a specific target device.

In regard to Claim 11, the examiner takes official notice that a page is a well-known display means for displaying data on a computer.

In regard to Claim 14, Perry and Silberbauer teach the method of Claim 1, but do not teach that the definition of an interaction includes a conditional that specifies a view to execute based on the value of an attribute in the attribute store. Laskoski, however, does teach storing an attribute value, where changing the value results in different views being displayed on the display means, hence the attribute specifies a given view based on its value (Column 4, lines 3-6 and lines 12-14). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where the definition of an interaction includes a conditional that specifies a view to execute based on the value of an attribute in the

attribute store, since this allows different views to be selected instantly by means of a parameter, instead of programming a new view each time it is desired.

10. Claim 12 is rejected under 35 U.S.C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Laskoski (U.S. Patent Number 5,428,554) and “Understanding JavaServer Pages Model 2 Architecture”, by Govind Seshadri, December 1999 (hereinafter Seshadri).

In regard to Claim 12, Perry, Silberbauer, and Laskoski teach the method of Claim 11, but do not teach that the page is a Java Server Page. Seshadri, however, does teach the benefit of using Java Server pages to display data (Page 1, lines 1-11). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to display output data on a display page, where the display page is a Java Server Page, since Java Server Pages allow users to use Java as a server-side scripting language.

11. Claim 13 is rejected under 35 U.S.C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Laskoski (U.S. Patent Number 5,428,554) and “What is ASP?” by Dave Beauchemin, 4/12/2000 (hereinafter Beauchemin).

In regard to Claim 13, Perry, Silberbauer, and Laskoski teach the method of Claim 11, but do not teach that the page is a Active Server Page. Beauchemin, however, does teach the benefit of using Active Server pages to display data (Page 1, lines 26-29). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to display output

data on a display page, where the display page is a Active Server Page, since Active Server Pages allow users to use VBScript, Jscript, and/or Perl and have it executed on a server.

12. Claim 17 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Eisenberg et al. (U.S. Patent Number 5,572,671).

In regard to Claim 17, Perry and Silberbauer teach the method of Claim 1, but do not teach when a command definition indicates whether the behavior of the command should be performed when an error is detected when performing the behavior of the command. Eisenberg, however, does teach indicating upon error of a program, whether the program should continue or halt (Column 1, lines 25-35). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where a command definition indicates whether the behavior of the command should be performed when an error is detected when performing the behavior of the command, since this allows a command to ignore an error or alert an error to a user.

13. Claim 18 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Hayashi (U.S. Patent Number 4,992,971).

In regard to Claim 18, Perry and Silberbauer teach the method of Claim 1, but do not teach a translator for preparing a command and for extracting attributes from the commands. Hayashi, however, does teach preparing a command and extracting parameters from the

command (Column 1, lines 21-35 and Column 2, lines 35-40). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 1, where a translator is provided for preparing a command and for extracting attributes from the commands, as taught by Hayashi, since this puts the command in a form that will allow it to be executed.

14. Claims 21, 22, 26, and 27 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Limondin et al. (U.S. Patent Number 6,226,783).

In regard to Claim 21, Perry teaches: providing a plurality of definitions of interactions of the program in the form of Visual Basic buttons that allows a user to interact with a program (Page 4, lines 7-10); each interaction having one or more command definitions in the form of Visual Basic code executed in response to the interaction (Page 5, lines 1-4); a command having an attribute and a behavior. Perry teaches the concept of a function that takes arguments (attributes) and performs a behavior based on the function code (Page 19). Perry teaches that an interaction with a program is identified by means of recognizing an event that has occurred in the program. Perry further teaches for each command of the identified interaction, using an input attribute and performing the behavior of the command with input attributes and extracting attributed values of the command by retrieving values of the output attributes of the command. Perry does not teach setting a value of the input attribute based on the value of an output attribute of a previously performed command. Limondin, however, does teach function ‘steps’ where the output of one step is used as the input of a next step (Column 4, lines 42-45). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to provide a

program with a plurality of interactions, where each interaction specifies a command with a behavior and attribute, where when an interaction is performed input variables are used to perform the behavior of the command to produce output variables, as taught by Perry, where the method further includes setting a value of the input attribute based on the value of an output attribute of a previously performed command, as taught by Limondin, since this allows functions to be more functionality-specific, and one function does not have to be constructed to encapsulate two functionally-independent operations.

In regard to Claim 22, Perry teaches receiving a request by way of an event which lets the program know that a user requests some function to be performed (Page 3, lines 11-12) and identifying the interaction associated with the received request by recognizing the user triggered event and performing the associated event procedure (Page 5, lines 6-9).

In regard to Claim 26, Limondin teaches setting the value of input attributes of a command with values from an output of a corresponding command (Column 4, lines 42-45).

In regard to Claim 27, Limondin teaches that the step function creates an object before all commands take place (Column 10, lines 34-40).

15. Claim 23 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Limondin et al. (U.S. Patent Number 6,226,783) and further in view of Nock (U.S. Patent Number 6,038,565).

In regard to Claim 23, Perry and Limondin teach the method of Claim 21, and Limondin further teaches an object implementing the command (Figure 7, item 650 and Column 4, lines 35-37), but neither teaches that the identifying of the input attribute of a command includes

Art Unit: 2122

identifying a set method of the object implementing the command. Nock, however, does teach using a 'set' method to identify and alter an object attribute (Column 5, lines 61-67 and Column 6, lines 1-3).

16. Claim 24 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Limondin et al. (U.S. Patent Number 6,226,783) and further in view of Thompson (U.S. Patent Number 5,872,977).

In regard to Claim 24, Perry and Limondin teach the method of Claim 21, but do not teach identifying the input attribute of a command includes identifying a data member of an object implementing the command. Thompson, however, does teach identifying a data member of an object as an input attribute to a command of the object (Column 8, lines 56-66). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 21, further where identifying the input attribute of a command includes identifying a data member of an object implementing the command, as taught by Thompson, since this allows objects to use private data members in object functions.

17. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Limondin et al. (U.S. Patent Number 6,226,783) and further in view of Silberbauer et al. (U.S. Patent Number 5,355,493).

In regard to Claim 25, Perry and Limondin teach the method of Claim 21, but do not teach storing an output attribute in an attribute store and setting an input attribute from a value in an attribute store. Silberbauer, however, does teach storing output attributes in a parameter

Art Unit: 2122

template (Column 4, lines 43-46) and setting values of parameters with values stored in an attribute store (Column 4, lines 24-25 and lines 38-41). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 21, where the method includes storing an output attribute in an attribute store and setting an input attribute from a value in an attribute store, since storing data in between operations allows other operations to take place before using the attributes in the store.

18. Claim 28 is rejected under 35 U.S.C. 103(a) as being unpatentable over Travostino et al. (U.S. Patent Number 6,564,325) in view of Shapiro et al. (U.S. Patent Number 5,257,363).

In regard to Claim 28, Travostino teaches receiving a list of functions to be invoked by a program (Column 3, lines 23-28) where the list of functions is used to generate source code (Column 3, lines 49-52). Travostino does not teach for each function identifying the names of formal parameters of the function and outputting an invoking of the function with names of actual parameters derived from the formal parameters. Shapiro, however, does teach invoking a function by supplying a list of actual parameters in place of formal parameters (Column 20, lines 11-17). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to receive a list of functions to be invoked by a program where the list of functions is used to generate source code, as taught by Travostino, where for each function identifying the names of formal parameters of the function and outputting an invoking of the function with names of actual parameters derived from the formal parameters, as taught by Shapiro, since this allows function definitions to be invoked with actual parameters at run time.



Art Unit: 2122

19. Claim 29 is rejected under 35 U.S.C. 103(a) as being unpatentable over Travostino et al. (U.S. Patent Number 6,564,325) in view of Shapiro et al. (U.S. Patent Number 5,257,363) and further in view of Stavran (U.S. Patent Number 6,012,149).

In regard to Claim 29, Travostino and Shapiro teach the method of Claim 28, but do not teach providing a signature of a function, including function name, and parameter types. Stavran, however, does teach representing a function by a signature with function name and parameter types (Column 5, lines 5-12). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 28, where the method further includes providing a signature of a function, including function name, and parameter types, as taught by Stavran, since this allows for perfect information in the generation of a function invocation.

20. Claim 30 is rejected under 35 U.S.C. 103(a) as being unpatentable over Travostino et al. (U.S. Patent Number 6,564,325) in view of Shapiro et al. (U.S. Patent Number 5,257,363) and further in view of Hildebrandt (U.S. Patent Number 5,535,390).

In regard to Claim 30, Travostino and Shapiro teach the method of Claim 28, but do not teach that a function has an associated object and including outputting source code to instantiate the object for each function. Hildebrandt, however, does teach associating an object with a function and including code that instantiates the object for the function (Column 4, lines 4-12). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 28, where a function has an associated object and including outputting source code to instantiate the object for each function, since this allows object functionality that exists only in the scope of the function.

Art Unit: 2122

21. Claim 31 is rejected under 35 U.S.C. 103(a) as being unpatentable over Travostino et al. (U.S. Patent Number 6,564,325) in view of Shapiro et al. (U.S. Patent Number 5,257,363) and further in view of Hildebrandt (U.S. Patent Number 5,535,390) and Nock (U.S. Patent Number 6,038,565).

In regard to Claim 31, Travostino, Shapiro, and Hildebrandt teach the method of Claim 30, but do not teach that the object has set and get methods for setting and getting values of parameters. Nock, however, does teach set and get methods of an object that retrieves and sets Object parameter values (Column 5, lines 61-67 and Column 6, lines 1-3). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 30, where the object has set and get methods for setting and getting values of parameters, since this allows the added security of only allowing an object to reference its own parameters.

22. Claim 32 is rejected under 35 U.S.C. 103(a) as being unpatentable over Travostino et al. (U.S. Patent Number 6,564,325) in view of Shapiro et al. (U.S. Patent Number 5,257,363) and further in view of Bland et al. (U.S. Patent Number 5,960,441).

In regard to Claim 32, Travostino and Shapiro teach the method of Claim 28, but do not teach receiving an alias for a name of a parameter. Bland, however, does teach using an alias for a name of a parameter (Column 9, lines 30-35). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 28, where an alias is used for the name of a parameter, since this allows for run-time binding of external data references.

23. Claims 36-38 are rejected under 35 U.S.C. 103(a) as being unpatentable over Limondin et al. (U.S. Patent Number 6,226,783) in view of "Sams Teach Yourself Visual Basic 6 in 21 Days" by Greg Perry, Sams, 1998 (hereinafter Perry).

In regard to Claim 36, Limondin teaches the computer readable medium of Claim 34, but does not teach that the program operates in a request-response environment, and a request specifies an interaction whose commands are to be performed. Perry, however, does teach receiving a request by way of an event which lets the program know that a user requests some function to be performed (Page 3, lines 11-12); identifying the interaction associated with the received request by recognizing the user triggered event and performing the associated event procedure (Page 5, lines 6-9); and providing a response generated by the view associated with the identified interaction. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to perform the method of Claim 34 where the program operates in a request-response environment, and a request specifies an interaction whose commands are to be performed, since this allows a dynamic program that responds to user requests.

In regard to Claim 37, the examiner takes official notice that associating a view with a program interaction to display a response is well known, and a user primarily interfaces with a computer program through a computer display.

In regard to Claim 38, Perry teaches altering a command button in the visual interface, hence adjusting a condition to determine a way to view the button (Page 4, lines 10-15).

24. Claim 39 is rejected under 35 U.S.C. 103(a) as being unpatentable over Limondin et al. (U.S. Patent Number 6,226,783) in view of Bland et al. (U.S. Patent Number 5,960,441).

In regard to Claim 39, Limondin teaches the medium of Claim 33, but does not teach including an alias for an attribute of a command. Bland, however, does teach using an alias for an attribute (Column 9, lines 30-35). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build a medium of Claim 33, where an alias is used for an attribute, since this allows for run-time binding of external data references.

25. Claim 40 is rejected under 35 U.S.C. 103(a) as being unpatentable over Limondin et al. (U.S. Patent Number 6,226,783).

In regard to Claim 40, the examiner takes official notice that using a constant for an attribute of a command is obvious, since this allows the command to use the constant to perform different meaningful calculations, without having to re-type a value of the constant every time a calculation is made.

26. Claim 41 is rejected under 35 U.S.C. 103(a) as being unpatentable over Limondin et al. (U.S. Patent Number 6,226,783) in view of Eisenberg et al. (U.S. Patent Number 5,572,671).

In regard to Claim 41, Limondin teaches the medium of Claim 33, but do not teach when a command definition indicates whether the behavior of the command should be performed when an error is detected when performing the behavior of the command. Eisenberg, however, does teach indicating upon error of a program, whether the program should continue or halt (Column 1, lines 25-35). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the medium of Claim 33, where a command definition indicates whether the behavior of the command should be performed when an error is detected when performing the behavior of the command, since this allows a command to ignore an error or alert an error to a user.

Art Unit: 2122

27. Claim 6, 7, and 45 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Limondin et al. (U.S. Patent Number 6,226,783).

In regard to Claim 6, Perry and Silberbauer teach the method of Claim 1, but do not teach that the command definition defines an object class and including instantiating an object class before setting the values of the input attributes of the command. Limondin, however, does teach defining an object class associated with the command (Figure 7, item 650). It is obvious that the class is instantiated before setting values of the command, since the command exists as part of the instantiation of the object (Column 4, lines 35-37 and Column 3, lines 3-4).

In regard to Claim 7, the examiner takes official notice that a Java Bean is a well-known object component and using a Java Bean allows for the portability and platform-independence of the Java language.

In regard to Claim 45, Perry and Silberbauer teach the system of Claim 42, but do not teach that setting an input attribute value occurs when a corresponding output attribute is retrieved. Limondin, however, does teach function ‘steps’ where the output of one step is used as the input of a next step (Column 4, lines 42-45). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the system of Claim 42, where setting an input attribute value occurs when a corresponding output attribute is retrieved, since this allows functions to be more functionality-specific, and one function does not have to be constructed to encapsulate two functionally-independent operations.

28. Claims 46 and 47 are rejected under 35 U.S.C. 103(a) as being unpatentable over “Sams Teach Yourself Visual Basic 6 in 21 Days” by Greg Perry, Sams, 1998 (hereinafter Perry) in view of Silberbauer et al. (U.S. Patent Number 5,355,493) and further in view of Nock (U.S. Patent Number 6,038,565) and Thompson (U.S. Patent Number 5,872,977).

In regard to Claim 46, Perry and Silberbauer teach the system of Claim 42, but do not teach that command have objects with get and set methods and wherein the means for preparing a command includes after identifying the input attribute, invoking a get method of an object to retrieve an attribute value. Nock, however, does teach get and set commands associated with objects for retrieving and setting values of parameters (Column 5, lines 61-67 and Column 6, lines 1-3). Nock does not teach preparing a command by invoking a get method of an object to retrieve an input attribute value. Thompson, however, does teach using an object command to retrieve an input for a program (Column 8, lines 56-66). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the system of Claim 42, where that command have objects with get and set methods, as taught by Nock, and wherein the means for preparing a command includes after identifying the input attribute, invoking a get method of an object to retrieve an attribute value, as taught by Thompson, since this allows better security of object variables by only allowing access to the variables by the object itself.

In regard to Claim 47, Nock teaches using a set command to set the value of an attribute value (Column 5, lines 61-67 and Column 6, lines 1-3).

29. Claims 49, 51, 53, and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Silberbauer et al. (U.S. Patent Number 5,355,493) in view of Limondin et al. (U.S. Patent Number 6,226,783).

In regard to Claim 49, Silberbauer teaches the medium of Claim 48, but does not teach that commands are organized into interactions. Limondin, however, teaches that each step has a list of commands associated with it, where steps are operations that interact with graphics (Column 3, lines 3-4). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to construct the medium of Claim 48, where commands are organized into interactions, since this allows a user to perform certain functions by interacting with a computer program.

In regard to Claim 51, Limondin teaches that each command has an object with a perform method (Figure 7, item 650).

In regard to Claim 53, the examiner takes official notice that an assignment statement in command code assigns values to attributes.

In regard to Claim 55, Silberbauer and Limondin teach the medium of Claim 48, and further teach that commands are organized into named interactions (Figure 5, item 200 and Figure 6, item 650) and that commands of interactions are performed when a request specifying the name of the interaction is received (Column 12, lines 46-55).

30. Claim 50 is rejected under 35 U.S.C. 103(a) as being unpatentable over Silberbauer et al. (U.S. Patent Number 5,355,493) in view of Limondin et al. (U.S. Patent Number 6,226,783) and further in view of Nock (U.S. Patent Number 6,038,565).

In regard to Claim 50, Silberbauer and Limondin teach the medium of Claim 48, and Limondin further teaches that each command has an object (Figure 7, item 650). Neither Silberbauer nor Limondin teach that the object has methods for setting and getting values of attributes of the object. Nock, however, does teach set and get methods of an object that retrieves

Art Unit: 2122

and sets Object parameter values (Column 5, lines 61-67 and Column 6, lines 1-3). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to construct the medium of Claim 48, where each command has an object, as taught by Silberbauer and Limondin, where the object has methods for setting and getting values of attributes of the object, as taught by Nock, since this allows the added security of only allowing an object to reference its own parameters.

31. Claim 52 is rejected under 35 U.S.C. 103(a) as being unpatentable over Silberbauer et al. (U.S. Patent Number 5,355,493) in view of Limondin et al. (U.S. Patent Number 6,226,783) and further in view of Bland et al. (U.S. Patent Number 5,960,441).

In regard to Claim 52, Silberbauer and Limondin teach the medium of Claim 48, but do not teach that the command includes an alias for an attribute. Bland, however, does teach using an alias for a name of a parameter (Column 9, lines 30-35). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to construct the medium of Claim 48, where an alias is used for the name of a parameter, since this allows for run-time binding of external data references.

32. Claim 54 is rejected under 35 U.S.C. 103(a) as being unpatentable over Silberbauer et al. (U.S. Patent Number 5,355,493) in view of Holloway et al. (U.S. Patent Number 6,636,855).

In regard to Claim 54, Silberbauer teaches the medium of Claim 48, but does not teach that the command has a translator for translating a value before setting an input attribute variable. Holloway, however, does teach translating a data variable before it is used as an input parameter (Column 5, 54-59). Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to build the medium of Claim 48, where the command has a translator



Art Unit: 2122

for translating a value before setting an input attribute variable, since this allows variables of different types from being used in a command that requires a given type input.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kenneth A Gross whose telephone number is (703) 305-0542.

The examiner can normally be reached on Mon-Fri 7:30-5.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tuan Q Dam can be reached on (703) 305-4552. The fax phone number for the organization where this application or proceeding is assigned is (703) 872-9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the receptionist whose telephone number is (703) 305-3900.

KAG



**ANTONY NGUYEN-BA  
PRIMARY EXAMINER**